A Natural Deduction Environment for Matita

Claudio Sacerdoti Coen <sacerdot@cs.unibo.it>

University of Bologna

12/07/2009

Outline

Outline

The Context

Bad news:

We elected Berlusconi (again!) . . .

The Context

Bad news:

We elected Berlusconi (again!) . . . hence yet another change in the undergraduate curricula

Good news:

We now teach a first semester, first year course in mathematical logic

The Problem

Contents:
A traditional CS-style course in logic (natural deduction and/or sequent calculus, . . .)

A basic course in interactive theorem proving (proofs in declarative language of propositional calculus meta-theory, e.g. the duality theorem or the Shannon expansion theorem)

Problem:
Not enough time to teach two different tools: need for

- A learning tool for natural deduction in Matita
- Smooth transition from ND to declarative language

Requirements:

- same information he would write on paper
  prevent inference
- allow incorrect derivation trees
  prevent proof-checking
- errors are highlighted, but not critical
  prevent proof-checking
- graphic display of the derivation tree (for oversized trees)
  trees not currently supported
- quick (batch) correction of exercises (for the teacher)
  re-enable proof-checking
- textual syntax (to smooth the transition)
- help for syntax learning and hasting the input phase
  a palette

Implementation:

Standard (bad) solutions:

- External UI, Matita as a service
  no smooth transition
- New component/plugin for Matita
  difficult to implement and to maintain, bad integration, boring

Proposed solution:

- "Implement" the system IN Matita, by exploiting the already available MKM technologies
- NO AD-HOC CHANGE to the code of Matita
  No change at all but for palettes!
- It works (quite surprisingly!)

Outline

MKM technologies in Matita

Matita is currently the most MKM-friendly ITP:

- management of a Web-distributed, inconsistent, library
- advanced indexing/searching on the global library
- XML technologies
- three levels of mathematical representation:
  - semantics (CIC)
  - content (OMDoc + MathML)
  - presentation (BoxML + MathML)
- MathML Presentation based UI
- unconstrained ambiguous notation and session profiling to aid ranking and solving ambiguities

Derivation tree rendering: ≡ MathML

“graphic display of the derivation tree (for oversized trees)”
“errors are highlighted”

MathML Presentation is quite good:

- No support for derivation trees
  (badly) approximated using fractions
- `maction` to expand/collapse sub-trees when too large
- `maction` to hide/show available hypotheses in sub-trees
- `mattr` to highlight errors
  how to detect them?

Incorrect input: semantic ambiguity

"allow incorrect derivation trees" *"errors are not critical"*
"batch correction of exercises (for the teacher)"

Three levels of representation:

- (MathML) Presentation; in many-to-many relation with
- (OMDoc + MathML) Content; in one(many)-to-many relation with
- (CIC) Semantics

Parallel markup between Semantics and Presentation:

(XSLT) transformations:	Semantics	$\longrightarrow$	Presentation.
Disambiguation:	Presentation	$\hookrightarrow$	Semantics

Incorrect input: semantic ambiguity

"allow incorrect derivation trees" *"errors are not critical"*
"batch correction of exercises (for the teacher)"

PARSING + DISAMBIGUATION

One presentation: $\dfrac{\begin{array}{cc}\vdots & \vdots \\ \mathcal{A} & \mathcal{B}\end{array}}{A \wedge B}\,\wedge_i$

One content: $\mathrm{AND}_i\ A\ B \quad \begin{array}{cc}\vdots & \vdots \\ \mathcal{A} & \mathcal{B}\end{array}$

Two (one) semantics: $\left(\mathrm{And}_i\ A\ B \quad \begin{array}{cc}\overset{A}{\vdots} & \overset{B}{\vdots} \\ A & B\end{array}\right) : A \wedge B$

$\left(\mathrm{And}_i^x\ A\ B\ \mathcal{A}\ \mathcal{B} \quad \begin{array}{cc}\overset{\mathcal{A}}{\vdots} & \overset{\mathcal{B}}{\vdots} \\ \mathcal{A} & \mathcal{B}\end{array}\right) : A \wedge B$

Incorrect input: semantic ambiguity

"allow incorrect derivation trees" *"errors are not critical"*
"batch correction of exercises (for the teacher)"

PRINTING

Two semantics: $(\mathrm{And}_i\ A\ B\ \overset{A}{\vdots}\ A\ \overset{B}{\vdots}\ B\ ) : A \wedge B$

$(\mathrm{And}_i^x\ A\ B\ \mathcal{A}\ \mathcal{B}\ \overset{\mathcal{A}}{\vdots}\ \mathcal{A}\ \overset{\mathcal{B}}{\vdots}\ \mathcal{B}\ ) : A \wedge B$

Two (one) contents: $\mathrm{AND}_i\ A\ B\ \vdots\ \mathcal{A}\ \vdots\ \mathcal{B}$ $\mathrm{AND}_i^x\ A\ B\ \vdots\ \mathcal{A}\ \vdots\ \mathcal{B}$

Two presentations: $\dfrac{\vdots\ \mathcal{A}\quad \vdots\ \mathcal{B}}{A \wedge B}\wedge_i$ $\dfrac{\vdots\ \mathcal{A}\quad \vdots\ \mathcal{B}}{A \wedge B}\wedge_i$

Outline

Outline

Practical results

A reasonable natural deduction environment for Matita built with NO lines of OCaml code (860 lines of definitions and notational declarations in Matita).

(MKM) Technologies used:

- MathML Presentation notation
- Multiple representations
- Disambiguation

Lessons learned

- Generic UI extensions (MathML, SVG, *ML, etc.) pay more than ad-hoc extensions
- Three levels of representation:
  Presentation, Content, Semantics
- Errors are just a form of semantics
  they can be internalized
- Disambiguation may subsume error recovery